Planning of the protoDUNE Prompt Processing System

M.Potekhin, B. Viren

March 18, 2017

Abstract

The purpose of this document is to inform the leadership of the Single-Phase protoDUNE experiment (NP04) about the scope, deliverables, schedules, interfaces and other crucial characteristics of the prompt processing (PP).

Contents

1 Overview

It is important to monitor the data being produced in the Single-Phase protoDUNE experiment in a way that allows fast identification and correction of any problems. Monitoring algorithms can require large volumes of data (high bandwidth) and/or large CPU resources. Algorithms that require both extremes may require more hardware than can be made available or may not return results quickly enough to be actionable. Within the spectrum of possibile CPU vs bandwidth strategies there are three categories of practical interest:

OM *Online Monitoring*: fast algorithms consuming large fractions of raw data and running within the DAQ computing environment.

PP *Prompt Processing*: relatively long-running algorithms consuming small subsamples of the raw data and running in a conventional computing environment with a workload management system optimized for promptness.

FP *Full Production*: any excess noise filtering, signal processing, activity imaging, object formation, pattern recognition and other reconstruction.

These three categories are naturally overlapping in terms of which algorithms can or will run in each. To efficiently use our limited software development effort it is important to find ways to create the needed algorithms in a way that allow their reuse in the three contexts. Because the code for all three categories are ultimately based on *art* this reuse can be achieved with relatively modest effort.[1]

In particular, this should allow OM and PP groups flexibility in terms of easily moving algorithms from one environment to another as we better understand our hardware resources and monitoring needs. This balance will likely be struck after we have developed and benchmarked the monitoring algorithms. It is expected that these benchmarks will be evaluated and algorithms selected in order that each category of jobs satisfies the very rough requirements illustrated in Table 1.

	triggers processed	avail CPU	result latency
OM:	$\sim 10\%$	tens	1 min
PP:	$\sim 1\%$	hundreds	10 min
FP:	100%	thousands	months

Table 1: Qualitative comparison and rough expectations for coverage, resources and performance of online monitoring (OM), prompt processing (PP) and full production (FP) categories of jobs.

With the above in mind, this note focuses on the unique parts that make up PP. It also describes ways in which the PP group intends to work with the OM and FP groups in order to best make use of our limited and shared effort in the collaboration.

2 The Scope of Prompt Processing

Prompt Processing can be conceptualized as consisting of (a) payload jobs necessary to fulfill its function, and (b) the infrastructure that makes execution of these jobs possible in an optimal

[1] The authors recommend that algorithm developers embrace the new "tool" feature provided by the latest version of *art*.

manner. It is important to ensure portability and interoperability of software created in the OM, PP and FP domains, especially since they are bound to use functionally similar or identical units of software. It is expected therefore that writing resusable payload jobs will be *a collaborative effort* involving members of the DRA and other groups, and their scope is discussed in Sec. 4.

The part of responsibility of the Prompt Processing group that is “internal” to it is the development of the *protoDUNE Prompt Processing System* (p3s) and it is the group’s main deliverable. The core of p3s is a specialized job management system which provides latency assurances for job completion using a fixed amount of computing resources at the expense of limiting the data coverage. This is opposite from traditional job management systems that emphasize processing 100% of all input data and effectively do not limit the duration of that processing. The Prompt Processing System also includes an array of User Interface and data visualization tools. The scope of p3s is described further in Sec. 3.

Finally, guidance on hardware requirements are given in Sec. 5 followed by a timeline and milestones in Sec. 6, a summary of resources in Sec. 7 and ending with a risk assessment in Sec. 8.

3 p3s

The *protoDUNE Prompt Processing System* (p3s) is primarily a platform for automated processing of data streams which aims to guarantee an agreed upon latency of producing results, as opposed to data coverage or throughput typical of a common Workload Management System. It is an infrastructure element and does not, as a deliverable, include the actual software for its computational payloads (which are discussed in Sec. 4).

A related but separate part of the PP deliverable is the p3s *visualization system*. This system is intended to display graphical (eg, histograms, plots, event displays) and other summary results produced by the payload jobs run by p3s. This portion of the system is still conceptual and there is hope that it can be realized using the same system needed for presenting OM results. The remainder of this section will make reference to the visualization system but is focused on the job management system.

3.1 Input, Intermediate and Output Files

From the point of view of raw data flow, the boundary between online and offline scope is the Online Buffer which is a storage system attached to DAQ computers. It exists primarily to assure continued DAQ operation in the event of an outage of the link to the CERN network. The “protoDUNE/SP Data Scenarios” spreadsheet [1] describes expected running conditions and provides estimates of data volumes and rates relevant to this boundary.

All raw files written by the DAQ to the buffer will be transferred to CERN EOS using the Fermilab file transfer system (F-FTS) [2,3]. F-FTS takes responsibility for purging any and all files it is given and can do so in a configurable manner.

The p3s payload jobs are expected to ingest a small percentage ($\lesssim$1%) of raw data files. This number is a very rough guess based on expected CPU requirements of the algorithms, amount of available CPU and amount of data to provide meaningful feedback about the data quality.

All files consumed and produced by p3s jobs are assumed to be served via XRootD [4]. This allows flexibility in ingesting the initial raw data files from one or both of the two possible sources:

- the Online Buffer
- CERN EOS [5]

The Online Buffer, if accessed at all, would only be used as a source of raw data files. Data streaming capability inherent in XRootD for many use cases can greatly limit the amount of data that must be transferred in the case where only a single trigger or even a subset of fragments of one trigger are needed from a raw data file to satisfy a given payload job. Jobs which do not have the capability or requirement to stream data must nevertheless provide a way to stage files in to and out from local storage. This is expected to be accomplished by a script which can likely be shared by a variety of core payload jobs.

The p3s will perform basics checks for successful job completion and existence of expected output files and purge intermediate data. As needed, some portion of the files produced by p3s jobs will be archived to mass storage and made available for distribution to the collaboration.

Final output files produced by these jobs are also presented to end users by the p3s *visualization system* in a number of ways as described below. The final output files can be categorized in the following way

graphics files in PNG or SVG format representing histograms, plots, event displays, etc.

summary files in JSON format provides ancillary information for graphics files including captions or other information that may be directly rendered by a user interface.

reference files which should be retained for browsing by the user and which may include logs or intermediate results.

The two end-user interfaces that p3s will provide are:

web pages a web server which allows browsing of all results, and include features such as refreshing of specific time critical results

notifications a system that interprets *summary* files to provide alarm notification via email, mobile push, or other means.

As mentioned above, the hope is that the visualization needs of the OM and PP results can be satisfied by a single system. The PP group will work with OM and others to achieve that.

3.2 Required Capabilities

The p3s must have capabilties to support categories of processing outlined in Sec. 4. The term "task" is used here to designate a set of related jobs, for a example a processing chain where each job consumes data produced by its predecessor. Jobs can be used to add a wide range of functionalty to the system, e.g. copy a fraction of data to mass storage if necessary or produce a set of PNG files based on intermediate ROOT files, or perhaps generate an alarm if a certain metric falls out of prescribed range.

The p3s supports description of tasks as DAGs and also supports task templates for ease of operation. The first job in the chain reads data from the Online Buffer or EOS. In the limiting case, a task may contain just one payload job; there is no set limit on the number of jobs in the DAG or its topology. Submission of individual *ad hoc* jobs by users is also supported.

Job and Task Management

- automatic generation of tasks upon arrival of fresh data to the buffer

- distribution of jobs to processing resources assigned to p3s
- management of tasks i.e. orchestration of execution of jobs within a task according to the dependencies between jobs
- prioritization of tasks and jobs (during assignmenet to processing slots) including automated dynamic changes of priorities in order to ensure completion of critical tasks

User and System interfaces

- full remote control of the system by operators via appropriate interfaces
- interface to the Online Buffer and EOS in order to access the input data
- interface to F-FTS necessary to move a portion of p3s output to external sites and mass storage
- Web interface for task monitoring and debugging of p3s

Web Access to Results

- the data produced by p3s will be made available to the users via a Web service (e.g. a Web page populated with plots selected according to user-specified criteria)
- for the data which is preserved as data files (as opposed to visual products) there will be a catalog accessible via a browser/Web page
- this item should be shared with OM.

Notification of Exceptions

- summary files will be interpreted and checked for fields holding metrics which are subject to limits defined in p3s.
- when a calculated metric is outside of the limits a notification is sent to registered individuals.
- domain experts will have ability to modify the list of watched quantities and their limits.

3.3 p3s Components and Deliverables

The core of p3s is a Django-based [9] application written in Python. The deliverables in the systems category are as follows:

p3s server — a web service which supports the capabilities described in Sec. 3.2

user tools — such as CLI clients necessary to manipulate the state of the system, submit and manage tasks in the manual mode if necessary, and to perform general management and maintenance of the system.

visualization — a web application, ideally shared with OM, which presents graphical and other summary results that were produced by p3s payload jobs available to end users.

Web and DB — the (software) servers needed as a platform to support the items listed above. These can be based on a variety of products and offer a degree of interoperability.

4 Payload Jobs

4.1 The Role of DQM Stakeholders

The algorithms to be used in the p3s payload jobs span a wide range of expertise in a few domains. Some jobs will consist of algorithms originated in the OM group but may have to be moved to PP due to CPU constraints.. On the other end of the spectrum are jobs based on algorithms that will run as a part of the *full production* (FP) processing which if fully in the offline domain.

Because of the considerable breadth of the required expertise the PP group should not commit to writing the code for all payload jobs. It will however facilitate the work of other collaborators and ensure development of optimal interfaces to data and applications. In particular the group will closely coordinate with OM and FP groups to identify individuals that can develop the algorithms. It will also assist in the development to help it progress in ways that exploit the benefits of *art* and realize the flexibility described above.

In the end, the exact makeup of the p3s payload jobs that will run must be determined by the detector hardware and the final DAQ architecture, and by joint decision of OM, PP, DRA groups and any other DQM stakeholders. However they are developed, it is any given job will implement one or more of the following stages which are described more fully in [6]. The output of each stage is logically an input to the next although, due to p3s, there is the potential for parallelization of stages. This is addressed by the required capabilities of p3s as outlined in Sec. 3.2.

4.2 TPC Processing Categories

DAQ category consists of results requiring no data decompression and likely will be produced in OM and by p3s jobs. It might includes summaries of things like (compressed) fragment sizes, error flags, data rates.

ADC results require data decompression but no other substantial CPU and will consist of summary of ADC-level data. Examples include mean/RMS values over time in various groupings and level of detail (ASIC, FEMB, RCE, APA etc).

FFT category consists of the application of discrete Fourier transforms (FFT) to ADC-level data primarily to understand any excess noise and its possible evolution.

SIG includes ADC mitigation, removal of any excess noise and the deconvolution of detector response functions in order to understand the signals related to actual activity in the detector.

RECO category includes application of any high-CPU algorithms for imaging and reconstructing the topology of activity in the detector and may include high level semantic event classifications.

4.3 Beam Instrumentation

It will be extremely desirable for protoDUNE during the data taking period to validate the trigger logic and other parameters related to the data coming from the Beam Instrumentation (BI) systems and their computing infrastructure elements. These data will not be fed into the DAQ system of the NP04 experiment so they will be captured separately and will have to be merged/integrated in a separate production step.

There will be several BI data streams such as coming from TOF, Cherenkov and fiber trackers. Some of these data will be "bundled" in packets correponding to the SPS cycle and will be available as a database entry for each cycle. The interface will be provided by CERN.

Because of the delay inherent in the flow of BI data as described above it cannot be optimally processed by the OM software, while full processing is not agile enough for the turnaround time required in DQM. For these reasons PP offers a unique opportunity to accomplish this part of DQM. Jobs in this category will obtain the BI data within minutes of it being produced and match it to the TPC and other protoDUNE subsystems.

4.4 Other Processing

There will be other protoDUNE components that may require DQM such as the photon detection and cosmic ray subsystems. Understanding the scope of required processing is expected to come from their respective workgroups.

5 Hardware

The p3s itself will require hardware to run its Web and database servers and to provide enough storage for intermediate and final output files as described in section 3.1. It is expected that this hardware will be provided by the CERN Neutrino Platform either from the pool of the machines which beloong to the existing [7] cluster, or procured separately if performance and scalability tests indicate such necessity.

The storage required for the output files is not currently known. It depends on the catalog of payload jobs that will be run and the desire of the collaboration for the retention of the results. The largest single type of result will be event displays. They will an amount of storage similar to the raw data from which they are produced. If one event display per minute is produced and retained indefinitely some few tens of TB will be required. The remanding class of results will require far less storage.

The hardware required to run the payload jobs themselves of course depends on the nature of their CPU needs and the fraction of triggers they consume. P3s follows a *pilot*-based paradigm and so is able to provide a uniform execution environment to its payload jobs over a variety of native hosting environments. This gives p3s the ability to scale elastically across different hosts (or facilties) as needed. Initial expectation is that DQM jobs will be run by p3s on nodes in the *neutdqm* partition of *neut* cluster. However it appears more optimal to run jobs in the Beam Instrumentation category on the *lxbatch* [8] facility because of its proximity to the CERN Central Services which include EOS and the database providing this type of data.

6 Timeline and Milestones

The items below include system deployment, functional and integration testing. Critical milestones are marked by bullets.

Feb'17: initial prototype — a working instance of p3s having most of required functionality, utilizing the "Django development server" and *sqlite* database back-end.

Mar'17: Apache/PostgreSQL — a continuously running instance of p3s deployed on Apache with PostgreSQL back-end, in a test setup at BNL. Resolution of concurrency issues.

Apr'17: initial CERN deployment — p3s deployed on the DQM (*neutdqm*) cluster.

May'17: XRootD — XRootD service on the DQM cluster with a functioning interface to EOS.

Jun'17: payload integration — art/LArSoft payloads confugured to run on p3s; system automated data driven generation and management of workflows

- DAQ vertical slice readiness

Jul'17: Visual and Data Products — Web server optimized for delivery of visual and data products produced by p3s.

Sep'17: p3s/DAQ integration testing —

- Full DAQ test readiness

Q4'17 — p3s stress and scalability testing, additional development

Q1'18 — continuous p3s operation with realistic workloads, running MC/Reco jobs in utility mode

Q2'18

- Full data taking readiness

7 Resources

7.1 Effort Levels

Development/engineering

- M.Potekhin (BNL) — 100% FTE
- B.Viren (BNL) — 10% FTE

System administration and support (working assumption)

- N.Benekos (CERN) — 50% FTE (TBD)
- G.Savage (FNAL) — 50% FTE (TBD)

The amount of effort needed for development of payload jobs is not considered in scope but see above sections for discussion of how the PP group will contribute and assist.

7.2 Matériel

A rough guess is that at a minimum p3s payload jobs will require O(100) cores during its operation. This is an elastic requirement. More available cores will allow more intensive processing over a larger percentage of the raw data while restricting the available hardware can be met by a scaling back. This scaling is automatically achieved by p3s.

An small number of adequately configured machines are needed for deployment of Web and DB servers to support the p3s function.

It is a working assumption at the time of writing that a significant portion of the *neut* cluster [7] will be assigned for p3s use as protoDUNE ramps up its operations in 2017 and into 2018. However and in any case the PP group relies on protoDUNE management to assist in finding suitable computing resources, especially in securing a guaranteed resource allocation on *lxbatch* if the Beam Instrumentation processing is to run on that facility.

8 Risk Assessment

Technologies used in p3s are not new, they are mature and tested in the field, which reduces overall implementation risk. The remaining risk factors are

- Integration of the Beam Instrumentation Data. The amount of development work necessary to access these data is not well understood. Optimal design of the merging process (e.g. to avoid doubling the data volume while writing out the merged data) can be a challenge.

- Insufficient allocation of hardware to p3s jobs can reduce the amount of data processed or the capability of the allowed algorithms to such an extent that the fraction of triggers or the types of features in the data which can be checked leads to a failure to identify problems in the detector.

- Lack of coordination and identification of individuals to develop the payload jobs which have sufficient coverage and performance may likewise lead to failure to identify problems in the detector.

- PP relies on the DAQ group to release updates to the "data access library" needed for unpacking raw data in a manner prompt enough so that p3s payload jobs can adapt, be rebuilt and deployed. Failures in this chain can lead to periods of time where detector data is unreadable and no data quality monitoring results are produced.

- If the small subsamples of the initial raw data is ingested from EOS then PP results depend on the successful and prompt transfer of raw data files from the Online Buffer to EOS by F-FTS. Any delay or outage will translate to late or missing PP results.

- If initial and ongoing expert system administration support is not identified for the computers running payload jobs and the servers running p3s web service and database then deployment of PP may be delayed or unreliable.

References

[1] DUNE DocDB 1086: *protoDUNE/SP data scenarios with full stream (spreadsheet)*
http://docs.dunescience.org:8080/cgi-bin/ShowDocument?docid=1086

[2] DUNE DocDB 1212: *Design of the Data Management System for the protoDUNE Experiment*
http://docs.dunescience.org:8080/cgi-bin/ShowDocument?docid=1212

[3] The Fermilab File Transfer System
http://cd-docdb.fnal.gov/cgi-bin/RetrieveFile?docid=5412&filename=datamanagement-changeprocedures.pdf&version=1

[4] XRootD
http://www.xrootd.org

[5] The CERN Exabyte Scale Storage
http://information-technology.web.cern.ch/services/eos-service

[6] DUNE DocDB 1811: *Prompt Processing System Requirements for the Single-Phase protoDUNE*
http://docs.dunescience.org:8080/cgi-bin/ShowDocument?docid=1811

[7] Neutrino Computing Cluster at CERN
https://twiki.cern.ch/twiki/bin/view/CENF/NeutrinoClusterCERN

[8] The CERN batch computing service
http://information-technology.web.cern.ch/services/batch

[9] Django
https://docs.djangoproject.com/en/1.10/